Mục lục

[Chương 1: Tổng quan đề tài 3](#_Toc10136)

[1.1 Giới thiệu đề tài 3](#_Toc10137)

[1.2 Phân loại có giám sát 3](#_Toc10138)

[1.3 Mục tiêu báo cáo 4](#_Toc10139)

[Chương 2: Nền tảng lý thuyết 4](#_Toc10140)

[2.1 Tìm hiểu về CNN 4](#_Toc10141)

[2.1.1 Mô hình neural network 5](#_Toc10142)

[2.1.2 Convolutional Neural Network 6](#_Toc10143)

[Convolution Layer 6](#_Toc10144)

[Pooling layer 7](#_Toc10145)

[Fully connected layer 8](#_Toc10146)

[2.2 Hàm sigmoid 8](#_Toc10147)

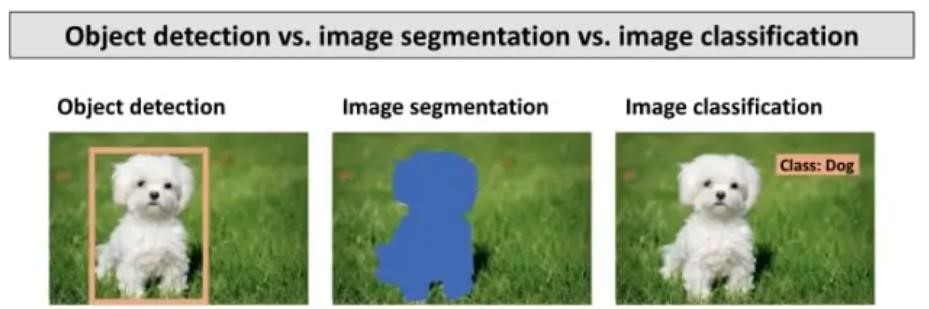
[2.3 Hàm Relu 10](#_Toc10148)

[Chương 3: Tập huấn dữ liệu và thực nghiệm 11](#_Toc10149)

# Chương 1: Tổng quan đề tài

## 1.1 Giới thiệu đề tài

Trong những năm qua, Thị giác máy tính là một trong những lĩnh vực rất nổi bật của khoa học máy tính và được ứng dụng rất thành công trong tự động hoá và robot. Phân loại hình ảnh (Image classification) hay Nhận dạng hình ảnh (Image recognition) là một trong những tác vụ của thị giác máy tính, ở đó thuật toán xem xét và dán nhãn cho hình ảnh từ một tập danh mục được xác định và đào tạo trước.



Hình 1:Sự khác biệt giữa phát hiện đối tượng, phân đoạn ảnh và phân loại ảnh.

Ví dụ, với một tập các hình ảnh, mỗi hình ảnh mô tả một con mèo hoặc một con chó, thuật toán sẽ “quan sát” toàn bộ dữ liệu và dựa trên hình dạng, màu sắc để hình thành giả thuyết liên quan đến nội dung của ảnh. Kết quả thu được là từ tập dữ liệu ban đầu, các hình ảnh chó/mèo đã được phân loại một cách tự động.

Thực tế, thị giác góp phần tạo nên 80-85% nhận thức của con người về thế giới. Hàng ngày, mỗi người phải thực hiện phân loại trên bất kỳ dữ liệu hình ảnh nào mà chúng ta bắt gặp.

Do đó, mô phỏng nhiệm vụ phân loại với sự trợ giúp của mạng nơ-ron là một trong những ứng dụng đầu tiên của thị giác máy tính mà các nhà nghiên cứu nghĩ đến.

Có nhiều thuật toán khác nhau được ứng dụng trong việc phân loại hình ảnh. Các thuật toán này được chia thành hai nhóm chính là Học có giám sát (supervised learning) và Học không giám sát (unsupervised learning). Nhưng trong bài này chúng ta sẽ thực hiện thuật toán học có giám sát (supervised learning).

## 1.2 Phân loại có giám sát

Trong học máy có giám sát, thuật toán được huấn luyện trên một tập hình ảnh đã được dán nhãn. Từ dữ liệu mẫu này, thuật toán có thể trích xuất thông tin, phục vụ phân loại ngay cả những hình ảnh chưa từng nhìn thấy trước đó.

Xuyên suốt quá trình đào tạo, đặc điểm của ma trận hình ảnh sẽ được trích xuất dưới dạng dữ liệu quan trọng để đưa vào xử lý. Các đặc điểm này đại diện cho hình ảnh trong không gian chiều thấp (lower-dimensional feature space) và là cơ sở để thuật toán tiến hành phân loại.

Trong quá trình đánh giá, các đặc điểm của ảnh thử nghiệm được thu thập và tái phân loại với sự hỗ trợ của mạng thần kinh nhân tạo. Hệ thống lúc này đã có thể nhận biết các đặc điểm điển hình của mọi lớp hình ảnh mà nó được đào tạo.

Các phương pháp phân loại phổ biến dựa trên học có giám sát bao gồm:

* Support Vector Machines
* Decision Trees
* K Nearest Neighbors

Các mạng nơ-ron thường được sử dụng để phân loại hình ảnh có giám sát bao gồm AlexNet, ResNet, DenseNet và Inception.

Đối với phân loại có giám sát, việc dán nhãn dữ liệu đóng vai trò quan trọng. Độ chính xác của dữ liệu được dán nhãn quyết định phần lớn hiệu suất của mô hình học máy. Các thuật toán phân loại có giám sát có thể được chia thành hai mục nhỏ hơn dựa trên nhãn dữ liệu.

## 1.3 Mục tiêu báo cáo

* Tìm hiểu các phương pháp tiếp cận và thực nghiệm phân loại ảnh.
* Khảo sát và thực nghiệm phương pháp phân loại ảnh chó mèo dựa trên mạng tích chập CNN.

# Chương 2: Nền tảng lý thuyết

## 2.1 Tìm hiểu về CNN

### 2.1.1 Mô hình neural network

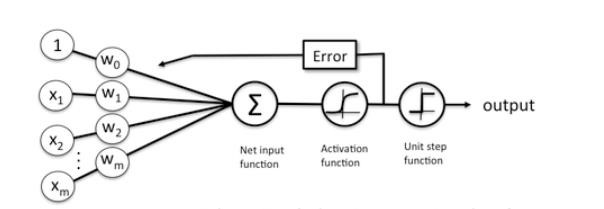
Mạng nơ-ron nhân tạo (Neural Network - NN) là một mô hình lập trình mô phỏng cách thức hoạt động của mạng nơ-ron thần kinh. Kết hợp với các kĩ thuật học sâu (Deep Learning - DL), mạng nơ-ron nhân tạo đang trở thành một công cụ rất mạnh mẽ mang lại hiệu quả tốt nhất cho nhiều bài toán khó như nhận dạng ảnh, giọng nói hay xử lý ngôn ngữ tự nhiên.

Lịch sử phát triển của mạng nơ-ron nhân tạo bắt đầu khi Warren McCulloch và Walter Pitts đã tạo ra một mô hình tính toán cho mạng nơ-ron dựa trên các thuật toán gọi là logic ngưỡng vào năm 1943. Tuy nhiên, để làm quen và hiểu được một số kiến thức cơ bản về mô hình mạng nơ-ron nhân tạp, chúng ta sẽ bàn đến hồi quy logictics - thứ có thể coi là mô hình neural network đơn giản nhất với chỉ input layer và output layer.

Giả sử chúng ta có bài toán dựa đoán khả năng nhận đơn giao hàng dựa trên khoảng cách và thời điểm cần giao trong ngày dựa trên dữ liệu đã cho trước. Từ đó ta có thể hiểu rằng với các bộ dữ liệu {x, y} cho sẵn với x có hai đặc trưng x1 và x2 lần lượt là khoảng cách và thời điểm cần giao trong ngày, chúng ta sẽ sử dụng các phương pháp để tìm được ước lượng . ŷ = f(w, x) sao cho sát với giá trị y nhất.

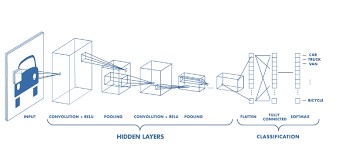
Thông thường, chúng ta thường sử dụng hàm f(w, x) = wTx để dễ tính toán, tuy nhiên đầu ra y là xác suất đơn hàng có được nhận hay không nên để đầu ra thỏa mãn được điều kiện có dạng tương tự xác suất tức là luôn có giá trị trong khoảng từ 0 đến 1 chúng ta thường sử dụng hàm logictics θ(z) = 1+1ⅇ−𝑧 với z = *f*(*w*,*x*) thường được gọi là hàm sigmod làm hàm activation.

Thông thường các bước tính toán của một mô hình mạng nơ-ron nhân tạo thường được thể hiện bằng một biểu đồ tính toán để có thể quan sát trực quan hơn. Dưới đây là một biểu đồ thể hiện cho bài toán phân loại dựa trên hồi quy logictics.



Hình 2: Sơ đồ hồi quy logictics.

### 2.1.2 Convolutional Neural Network



Hình 3. Sơ đồ CNN.

Như trình bày ở trên, Convolutional Neural Network là một trong những phương pháp chính khi sử dụng dữ liệu về ảnh. Kiến trúc mạng này xuất hiện do các phương pháp xử lý dữ liệu ảnh thường sử dụng giá trị của từng pixel. Vậy nên với một ảnh có giá trị kích thước 100x100 sử dụng kênh RGB ta có tổng cộng ta có 100 \* 100 \* 3 bằng 30000 nút ở lớp đầu vào. Điều đó kéo theo việc có một số lượng lớn weight và bias dẫn đến mạng nơ-ron trở nên quá đồ sộ, gây khó khăn cho việc tính toán. Hơn nữa, chúng ta có thể thấy rằng thông tin của các pixel thường chỉ chịu tác động bởi các pixel ngay gần nó, vậy nên việc bỏ qua một số nút ở tầng đầu vào trong mỗi lần huấn luyện sẽ không làm giảm độ chính xác của mô hình. Vậy nên người ta sử dụng cửa số tích chập nhằm giải quyết vấn đề số lượng tham số lớn mà vẫn trích xuất được đặc trưng của ảnh.

Về mặt kỹ thuật, trong mô hình học sâu CNN, mô hình ảnh đầu vào sẽ chuyển nó qua một loạt các lớp tích chập với các bộ lọc, sau đó đến lớp Pooling, rồi tiếp theo là các lớp được kết nối đầy đủ (FC — fully connected layers) và cuối cùng áp dụng hàm softmax để phân loại một đối tượng dựa trên giá trị xác suất trong khoản từ 0 đến 1.

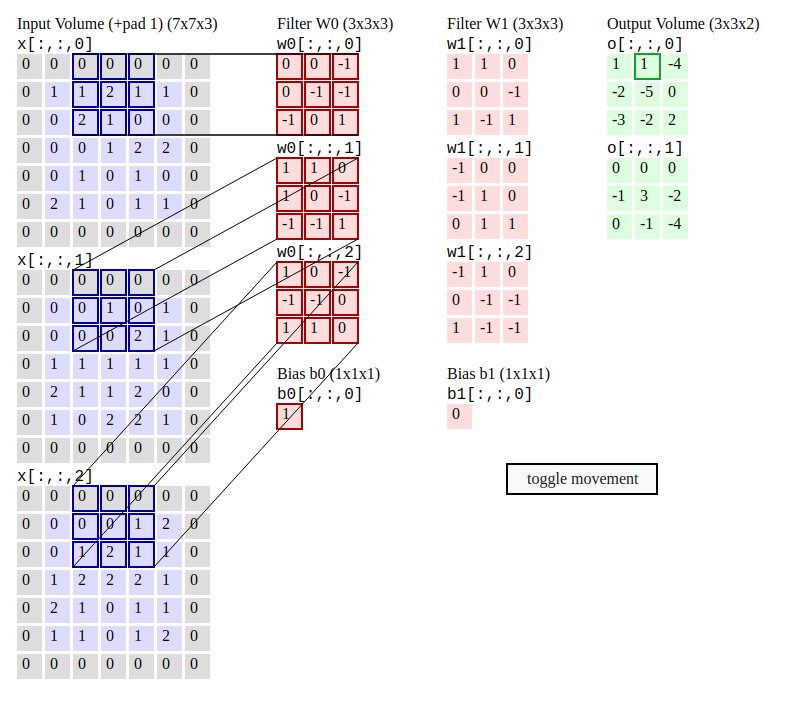
#### Convolution Layer

Convolution (lớp tích chập) là lớp đầu tiên trích xuất các đặc tính từ hình ảnh. Tham số lớp này bao gồm một tập hợp các bộ lọc có thể học được. Các bộ lọc đều nhỏ thường có kích cỡ hai chiều đầu tiên khoảng 3x3 hoặc 5x5, .... và có độ sâu bằng với độ sâu của đầu vào đầu vào. Bằng cách trượt dần bộ lọc theo chiều ngang và dọc trên ảnh, chúng thu được một Feature Map chứa các đặc trưng được trích xuất từ trên hình ảnh đầu vào.

Quá trình trượt các bộ lọc thường có các giá trị được quy định bao gồm:

* Padding: quy định bộ đệm của bộ lọc hay chính là phần màu xám được thêm vào ảnh
* Stride: quy định bước nhảy trong quá trình thực hiện.

Hình minh họa sau sẽ giúp chúng ta dễ tưởng tượng hơn về quá trình trên:

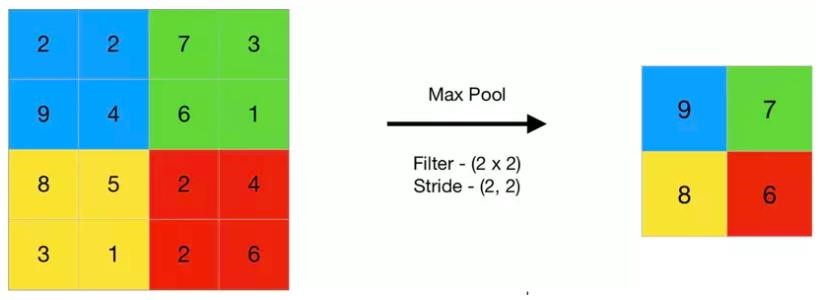


Hình 4: Convolution Layer.

Với mỗi kernel khác nhau ta sẽ học được những đặc trưng khác nhau của ảnh, nên trong mỗi convolutional layer ta sẽ dùng nhiều kernel để học được nhiều thuộc tính của ảnh. Vì mỗi kernel cho ra output là 1 matrix nên k kernel sẽ cho ra k output matrix. Ta kết hợp k output matrix này lại thành 1 tensor 3 chiều có chiều sâu k. Output của convolutional layer sẽ qua hàm activation function trước khi trở thành input của convolutional layer tiếp theo.

#### Pooling layer

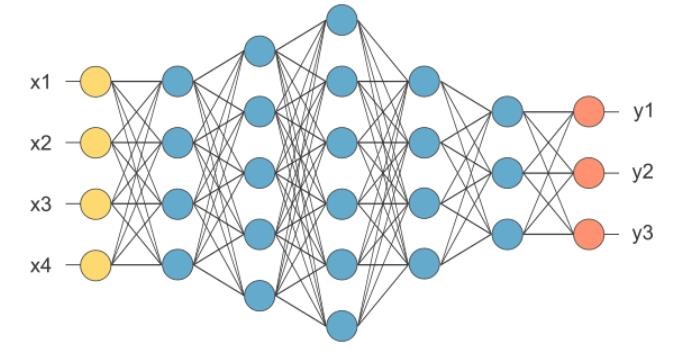
Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model. Trong quá trình này, quy tắc về stride và padding áp dụng như phép tính convolution trên ảnh.



Hình 5: Pooling layer.

#### Fully connected layer

Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh thì tensor của output của layer cuối cùng sẽ được là phẳng thành vector và đưa vào một lớp được kết nối như một mạng nơ-ron. Với FC layer được kết hợp với các tính năng lại với nhau để tạo ra một mô hình. Cuối cùng sử dụng softmax hoặc sigmoid để phân loại đầu ra.



Hình 6: Phân loại đầu ra.

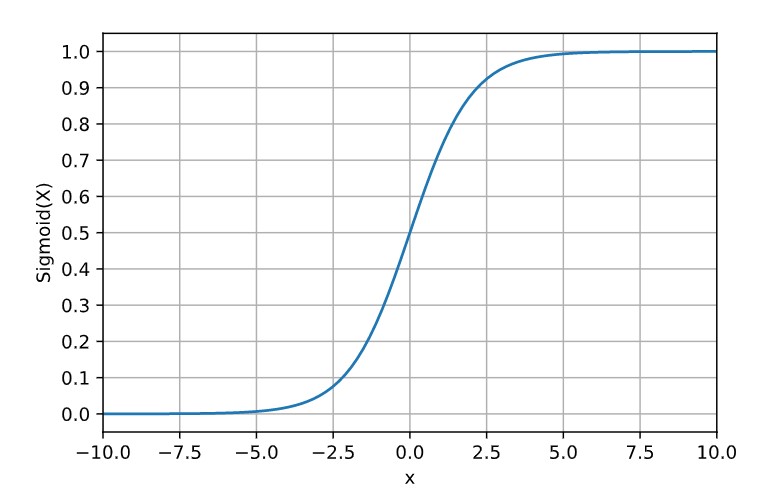
## 2.2 Hàm sigmoid

Công thức :

1

σ (x) = : 1+ⅇ−x

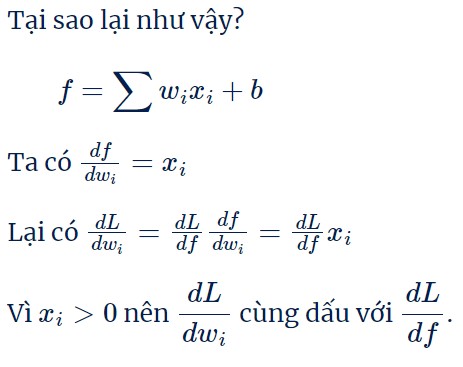
Phân tích



Hình 7: Đồ thị hàm sigmoid.

Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1) (xem đồ thị phía trên). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp. Tuy nhiên hiện nay hàm Sigmoid rất ít được dùng vì những nhược điểm sau:

* Hàm Sigmoid bão hào và triệt tiêu gradient: Một nhược điểm dễ nhận thấy là khi đầu vào có trị tuyệt đối lớn (rất âm hoặc rất dương), gradient của hàm số này sẽ rất gần với 0. Điều này đồng nghĩa với việc các hệ số tương ứng với unit đang xét sẽ gần như không được cập nhật (còn được gọi là *vanishing gradient*).
* Hàm Sigmoid không có trung tâm là 0 gây khó khăn cho việc hội tụ.



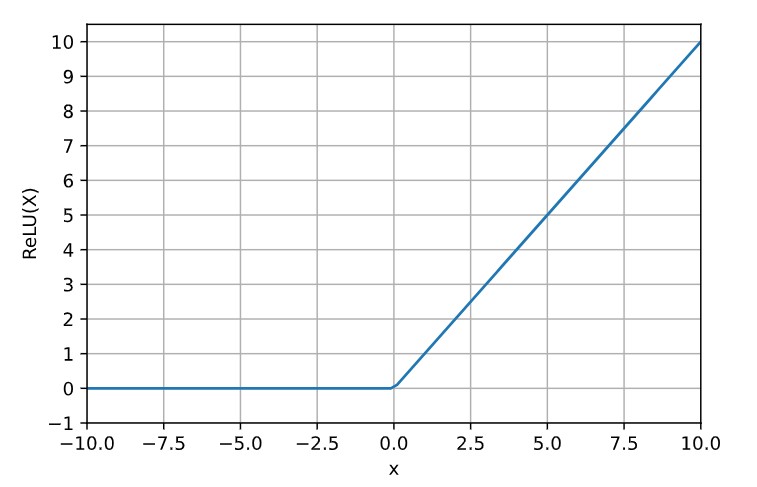
Rất may chúng ta có thể giải quyết vấn đề này bằng cách chuẩn hoá dữ liệu về dạng có trung tâm là 0 (zero-centered) với các thuật toán batch/layer normalization.

## 2.3 Hàm Relu

Công thức:

f (x) = max(0,x)

Phân tích



Hình 8: Đồ thị hàm Relu.

Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0. Nhìn vào công thức chúng ta dễ dàng hiểu được cách hoạt động của nó. Một số ưu điểm khá vượt trội của nó so với Sigmoid và Tanh:

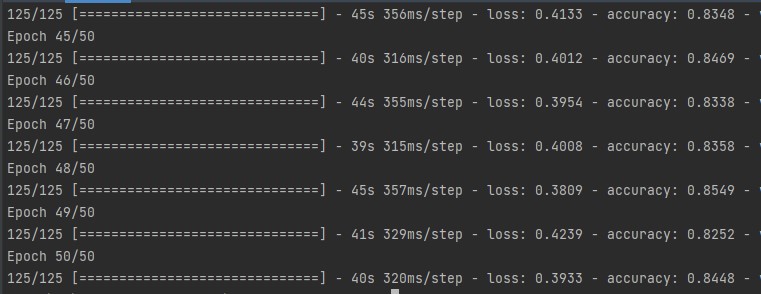
* Tốc độ hội tụ nhanh hơn hẳn. ReLU có tốc độ hội tụ nhanh gấp 6 lần Tanh. Điều này có thể do ReLU không bị bão hoà ở 2 đầu như Sigmoid và Tanh.
* Tính toán nhanh hơn. Tanh và Sigmoid sử dụng hàm exp và công thức phức tạp hơn ReLU rất nhiều do vậy sẽ tốn nhiều chi phí hơn để tính toán.

Tuy nhiên ReLU cũng có nhược điểm:

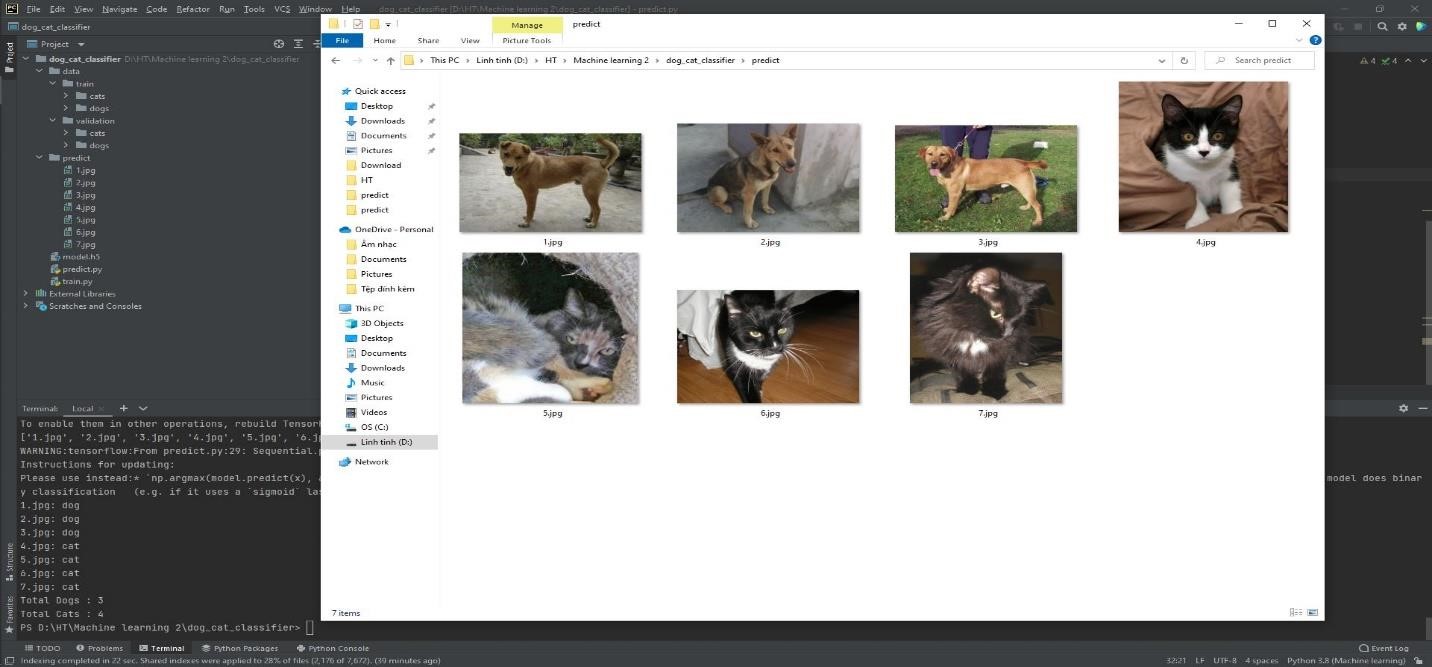
* Với các node có giá trị nhỏ hơn 0, qua ReLU activation sẽ thành 0, hiện tượng đấy gọi là “Dying ReLU“. Nếu các node bị chuyển thành 0 thì sẽ không có ý nghĩa với bước linear activation ở lớp tiếp theo và các hệ số tương ứng từ node đấy cũng không được cập nhật với gradient descent. => Leaky ReLU ra đời.
* Khi learning rate lớn, các trọng số (weights) có thể thay đổi theo cách làm tất cả neuron dừng việc cập nhật.

# Chương 3: Tập huấn dữ liệu và thực nghiệm

Kết quả:



Hình 9. Hiệu suất model đạt trên 82%.



Hình 10. Model nhận diện thành công một số ảnh.